UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

APLICACIÓN DE REALIDAD AUMENTADA PARA FAVORECER EL APRENDIZAJE AUTÓNOMO DE PERSONAS CON DIVERSIDAD FUNCIONAL INTELECTUAL

Ander Goig Fernández de Mendiola
Tutor: Juan Carlos Torrado Vidal
Ponente: Germán Montoro Manrique

JULIO 2018

APLICACIÓN DE REALIDAD AUMENTADA PARA FAVORECER EL APRENDIZAJE AUTÓNOMO DE PERSONAS CON DIVERSIDAD FUNCIONAL INTELECTUAL

AUTOR: Ander Goig Fernández de Mendiola
TUTOR: Juan Carlos Torrado Vidal

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2018

Resumen (castellano)

Ya hace mas de una década que convivimos con los *smartphones*, estos han cambiado la forma en la que nos comunicamos, la manera en la que accedemos a la información y, en general, nuestro estilo de vida. Desde un dispositivo no mucho mas grande que la palma de una mano, tenemos acceso a un sin fin de servicios.

Por otro lado, el culpable de esta revolución no es solo el *smartphone* en sí, si no tan bien, las aplicaciones móviles. Tras la inauguración en 2008, de la tienda de aplicaciones de Apple, la App Store, las aplicaciones se han instaurado, prácticamente, en todos los ámbitos de nuestra vida y se han convertido en casi indispensables.

Este Trabajo de Fin de Grado propone la realización de una aplicación móvil destinada al sector educativo. Orientada, principalmente, a alumnos con diversidad funcional intelectual como herramienta o instrumento de apoyo en la autoevaluación de fichas de ejercicios.

Para ello, la aplicación hará uso de técnicas de realidad aumentada. Esta tecnología se esta aplicando cada vez mas a los distintitos ámbitos de la vida cotidiana y permite visualizar un entorno físico real a través de un dispositivo, el cual permite mostrar una capa adicional de elementos virtuales.

Por lo tanto, mediante el uso de esta tecnología, se pretende reforzar el aprendizaje por descubrimiento de los alumnos, su interactividad y en definitiva su motivación.

En este documento se recogen las diferentes fases que se han llevado a cabo durante la realización del proyecto, el análisis del problema, el diseño de la solución, su implementación y las pruebas realizadas.

Abstract (English)

We have been living with smartphones for more than a decade now, they have changed the way we communicate, the way we access information and, in general, our whole lifestyle. From a device, not much larger than the palm of a hand, we have access to endless services.

On the other hand, the smartphone itself is not the only guilty, but also, the mobile applications. After the inauguration in 2008 of the Apple App Store, application have been established, practically, in all areas of our lives and have become almost indispensable.

This Bachelor Thesis proposes the implementation of a mobile application aimed at the educational field. Oriented, mainly, to students with intellectual functional diversity, as a support tool or instrument in the self-evaluation of exercise sheets.

To do that, the application will make use of augmented reality techniques. This technology is being applied more and more to different areas of day to day basis and allows you to visualize through a device, real physical environments with additional virtual elements on a higher layer.

Therefore, it is intended to reinforce, through the use of this technology, the learning by discovery of the students, their interactivity and ultimately, their motivation.

This document also includes the different phases that have been carried out during the realization of this project, like the analysis of the problem, the solution design, its implementation and the performed tests.

Palabras clave (castellano)

Realidad aumentada, Aplicación móvil, Educación, Android, Java

Keywords (inglés)

Augmented reality, Mobile app, Education, Android, Java

Agradecimientos

A mi familia, en especial a mis padres, por no dejar de apoyarme y hacer que todo esto sea posible.

INDICE DE CONTENIDOS

1 Introducción....1
1.1 Motivación....1
1.2 Objetivos....1
1.3 Organización de la memoria....1
2 Estado del arte....3
2.1 Soluciones existentes....3
2.1.1 Arloon....3
2.1.2 FETCH! Lunch Rush....4
2.1.3 Quiver Education....5
2.2 Conclusiones....6
3 Análisis....7
3.1 Análisis de requisitos....7
3.1.1 Requisitos funcionales....7
3.1.2 Requisitos no funcionales....9
3.2 Análisis de las soluciones....9
3.2.1 Soluciones para la identificación de fichas de ejercicios....10
3.2.2 Soluciones para la realidad aumentada....11
3.3 Solución propuesta....13
4 Diseño y desarrollo....15
4.1 Análisis de las herramientas....15
4.1.1 Entorno....15
4.1.2 Lenguajes....16
4.1.3 Librerías....16
4.2 Arquitectura software....18
4.2.1 Arquitectura en la realidad aumentada....19
4.2.2 Arquitectura en la gestión de fichas....20
4.3 Modelo de datos....21
5 Pruebas y resultados....23
5.1 Tipos de pruebas....23
5.2 Resultados....23
6 Conclusiones y trabajo futuro....25
6.1 Conclusiones....25
6.2 Trabajo futuro....25
Referencias....27
Glosario....29
Anexos....I
A Términos en la realidad aumentada....I
B Manual de usuario....I
C Maquetas de la aplicación....III
D Ejemplo de ficha de ejercicios....IX

INDICE DE FIGURAS

FIGURA 2-1: CAPTURA DE PANTALLA DE ARLOON GEOMETRY 4

FIGURA 2-2: MARCADOR DE AR DE ARLOON..... 4

FIGURA 2-3: MARCADOR DE AR DE FETCH! LUNCH RUSH 5

FIGURA 2-4: CAPTURA DE PANTALLA DE FETCH! LUNCH RUSH..... 5

FIGURA 2-5: CAPTURA DE PANTALLA DE QUIVER EDUCATION..... 6

FIGURA 4-1: ENTORNOS DE DESARROLLO INTEGRADO PARA ANDROID 15

FIGURA 4-2: MARCADOR AR DE LA APLICACIÓN 17

FIGURA 4-3: DIAGRAMAS DE CLASES SIMPLIFICADO 18

FIGURA 4-4: ARQUITECTURA DE LAS CLASES IMPLICADAS EN LA REALIDAD AUMENTADA 19

FIGURA 4-5: ARQUITECTURA DE LAS CLASES IMPLICADAS EN LA GESTIÓN DE FICHAS..... 20

FIGURA 4-6: MODELO DE DATOS 21

FIGURA 0-1: PANTALLA DE INICIO DE LA APLICACIÓN..... I

FIGURA 0-2: PANTALLA DE LA APLICACIÓN PARA LA CORRECCIÓN DE EJERCICIOS..... I

FIGURA 0-3: PANTALLA DE LA APLICACIÓN PARA LA CORRECCIÓN DE EJERCICIOS (CON AR)..... I

FIGURA 0-4: LISTA DE EJERCICIOS VACÍA I

FIGURA 0-5: TIPOS DE EJERCICIOS I

FIGURA 0-6: NUEVO EJERCICIO DE TIPO SUMA I

FIGURA 0-7: LISTA DE EJERCICIOS I

FIGURA 0-8: MAQUETA DE LA PANTALLA DE INICIO..... III

FIGURA 0-9: MAQUETA DE LA VISTA DE REALIDAD AUMENTADA..... V

FIGURA 0-10: MAQUETA DE LA GESTIÓN DE EJERCICIOS VII

FIGURA 0-11: EJEMPLO DE FICHA DE EJERCICIOS IX

1 Introducción

1.1 Motivación

Este proyecto busca aprovechar el uso de las Tecnologías de la Información y la Comunicación (TIC) para ayudar a alumnos con diversidad funcional intelectual a desarrollar habilidades y adquirir competencias dentro del ámbito educativo.

Para ello se aplicarán diversas herramientas de las TIC como el uso de aplicaciones móviles y el uso de nuevas tecnologías como la realidad aumentada.

1.2 Objetivos

El objetivo de este trabajo es la implementación de una aplicación móvil, dirigida principalmente al sector educativo y, en especial, a alumnos con diversidad funcional intelectual. La cual, hará uso de técnicas de realidad de aumentada para ayudarles a autoevaluar sus resultados tras la realización de diferentes fichas de ejercicios. Además, permitirá a los profesores crear, desde la propia aplicación, diferentes tipos de ejercicios.

Para ello se seguirán todas las fases de un desarrollo software. Se hará un estudio del dominio de la aplicación, se analizarán los requisitos de la aplicación en base a las necesidades de los usuarios, se examinarán las diferentes soluciones posibles y se diseñará, se implementará y se probará la solución.

1.3 Organización de la memoria

La memoria consta de los siguientes cinco capítulos, sin incluir la introducción:

- **Capítulo 2 -** Estado del arte
- **Capítulo 3 -** Análisis
- **Capítulo 4 -** Diseño y desarrollo
- **Capítulo 5 -** Pruebas y resultados
- **Capítulo 6 -** Conclusiones y trabajo futuro

2 Estado del arte

Para entender la importancia y el alcance de este proyecto es importante comprender el dominio en el que se centra la aplicación. Por ello, esta sección se centrará en el estudio y discusión de las herramientas, tecnologías y aplicaciones disponibles actualmente en relación con la realidad aumentada y más específicamente, aquellas dirigidas al aprendizaje o centradas, de alguna manera, como herramienta de apoyo en el aula.

2.1 Soluciones existentes

Si identificamos el dominio o nicho en el que se centra este proyecto, en seguida nos podemos dar cuenta que éste es bastante específico y poco genérico.

Si centramos la búsqueda de aplicaciones, en aquellas orientadas a personas con diversidad funcional intelectual, que se apoyen el aprendizaje autónomo y que, además, usen para ello tecnologías relativamente nuevas como la realidad aumentada, es difícil encontrar resultados relevantes con tales características.

Por otro lado, y aunque no estén dirigidas específicamente a personas con diversidad funcional intelectual, si que existen varios ejemplos de aplicaciones que hacen uso de la realidad aumentada de una forma didáctica, para así favorecer el aprendizaje y la enseñanza.

Es por ello por lo que se ha centrado el estudio en dichas aplicaciones. En las próximas secciones se muestran varios ejemplos.

2.1.1 Arloon

Arloon [3], en este caso, no es una aplicación en sí. Es una empresa, de origen valenciano, que se dedica precisamente a la creación de aplicaciones de contenido didáctico como un complemento a la enseñanza en el aula.

Entre sus aplicaciones, enfocadas a distintos tipos de materias (matemáticas, química, ciencias, etc.) y niveles educativos, existen varios ejemplos que hacen uso de la realidad aumentada.

Entre estos ejemplos están: **Arloon Geometry**, **Arloon Anatomy**, **Arloon Chemistry** o **Arloon Plants**. Las cuales se centran en la geometría, el cuerpo humano, la química y la botánica respectivamente.

Todas son aplicaciones de pago y multiplataforma, es decir, están disponibles tanto para Android como para iOS.

Estéticamente hablando, las cuatro son muy similares y mantienen un estilo muy parecido en cuanto a interfaz gráfica. La cual destaca por ser simple, intuitiva y con colores vivos, orientada un poco a un público infantil.

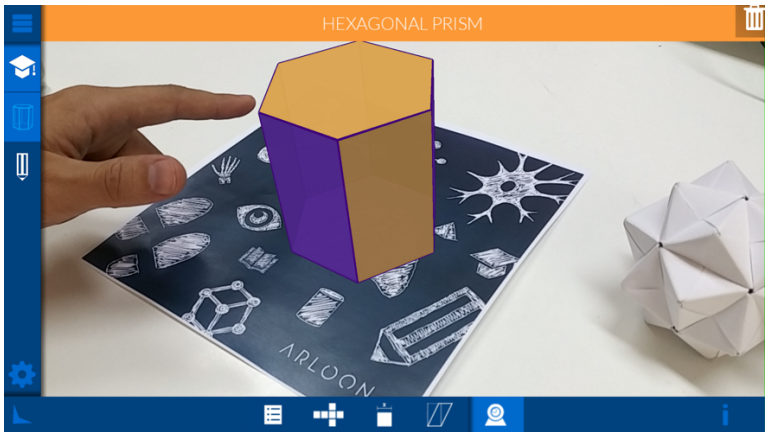

Figura 2-1: Captura de pantalla de Arloon Geometry

Centrándonos en la realidad aumentada, todas hacen uso de ella de alguna manera. En general, la utilizan para mostrar de una forma más visual e interactiva, diferentes modelos en tres dimensiones, en algunos casos animados, que varían dependiendo de la materia en la que se ubican. Como, por ejemplo, cuerpos geométricos, compuestos químicos o partes del cuerpo humano.

Figura 2-2: Marcador de AR de Arloon

La exposición de estos modelos 3D en realidad aumentada, se realiza en todos los casos mediante el uso de un marcador. Esto ayuda a la aplicación a calcular la posición y la orientación en la que debe ser mostrado dicho modelo [Figura 2-2].

2.1.2 FETCH! Lunch Rush

FETCH! Lunch Rush [4] es una aplicación disponible gratuitamente para dispositivos Android, iOS o Windows Phone. Se trata de un pequeño juego en el que los niños deben resolver sencillas operaciones matemáticas y en el que además se puntúan los aciertos y se valora la velocidad en la que se resuelven.

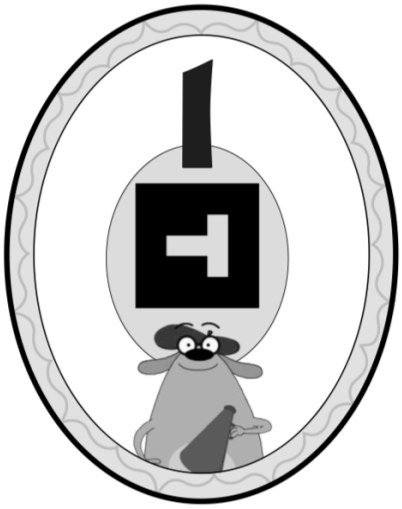

Figura 2-3: Marcador de AR de FETCH! Lunch Rush

Para ello se poseen una serie de cartas o marcadores en papel [Figura 2-3] donde cada una de ellas se corresponde con un número del 1 al 10. El ejercicio se resolverá apuntando con la cámara al marcador que corresponda en el que se muestran tantos "trozos de sushi" como indica el número [Figura 2-4].

Figura 2-4: Captura de pantalla de FETCH! Lunch Rush

2.1.3 Quiver Education

En este caso, **Quiver Education** [5] es una aplicación de dibujo de realidad aumentada, pero con un contenido mas enfocado al ámbito educativo. Es de pago (con descuentos a colegios) y esta disponible para dispositivos iOS o Android.

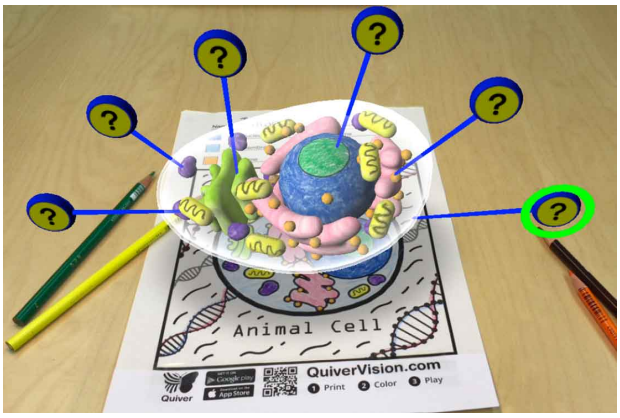

Figura 2-5: Captura de pantalla de Quiver Education

Las plantillas para dibujar son descargables y hay una gran variedad. Una vez coloreadas el usuario puede ver como sus dibujos "cobran vida" [Figura 2-5], pudiendo así visualizarlos en tres dimensiones, desde todas sus perspectivas, así como interactuando y aprendiendo con ellos.

2.2 Conclusiones

Se han mostrado varios ejemplos de aplicaciones de realidad aumentada orientadas a la educación y al aprendizaje de diferentes materias para así hacernos una idea y comprender que tipo de soluciones se pueden encontrar y como están enfocadas.

Existen algunas aplicaciones más además de las nombradas, pero debido a su similitud con las anteriores no se ha considerado relevante detallarlas. Han sido varios también los proyectos o aplicaciones que se han descartado debido a que ya no se encuentran disponibles actualmente o han sido descontinuados, como por ejemplo Aurasma [6].

Si bien es cierto que la realidad aumentada es una tecnología relativamente joven, es un mercado en auge y que no ha parado de crecer durante los últimos años. Uno puede encontrar cada vez más variedad de este tipo de aplicaciones, aun así, todavía es difícil encontrar una gran variedad que estén centradas específicamente en la educación.

3 Análisis

Esta sección describe la fase de análisis del proyecto. Se trata de una fase muy importante, ya que cometer errores en durante esta puede resultar en que el producto final no satisfaga las necesidades de los usuarios.

En ella se identifican los requisitos funcionales y no funcionales de la aplicación. Los cuales, han sido capturados a partir de las necesidades del cliente y de el estudio de otras aplicaciones similares en el mercado.

Por otro lado, se analizarán en profundidad algunos de los requisitos para tratar de entenderlos y encontrar las soluciones que mejor se adapten.

3.1 Análisis de requisitos

Los requisitos de la aplicación especificados a continuación se dividen en requisitos funcionales y requisitos no funcionales [1].

Los **requisitos funcionales** detallan los servicios que la aplicación debe proporcionar y cumplir, cómo debería comportarse en diferentes situaciones y como debería reaccionar ante diferentes entradas.

Por otro lado, los **requisitos no funcionales** describen las restricciones o condiciones en los servicios que ofrece el sistema, como, por ejemplo, restricciones de rendimiento, estabilidad, usabilidad, etc. Es decir, atributos de calidad.

El formato que siguen los requisitos será el siguiente:

<Identificador del requisito> *<Descripción del requisito>*
- *<Explicación mas detallada del requisito (en caso de que sea necesario)>*

3.1.1 Requisitos funcionales

Para una mejor comprensión, los **requisitos funcionales** presentados a continuación se han divido en dos subsistemas. Estos dos subsistemas se corresponden con los dos objetivos generales de la aplicación.

Por un lado, se encuentran los requisitos relacionados con la **gestión de fichas de ejercicios** y, por otro lado, los implicados en la **autoevaluación de los resultados mediante realidad aumentada**.

3.1.1.1 Requisitos en la gestión de fichas de ejercicios

RF01 Generar fichas de ejercicios.

- La aplicación permitirá a los profesores generar fichas de ejercicios. Los ejercicios consistirán en simples operaciones de matemáticas binarias (sumas, restas, multiplicaciones y divisiones).
- La aplicación presentará un formulario que el profesor rellenará con las características del ejercicio (tipo de operación matemática y operandos).

RF02 Guardar fichas de ejercicios.

- La aplicación permitirá almacenar en el dispositivo las fichas de ejercicios previamente generadas.

RF03 Listar fichas de ejercicios.

- La aplicación deberá mostrar en una lista la colección de fichas de ejercicios generadas y guardadas por los profesores.

RF04 Imprimir fichas de ejercicios.

- La aplicación permitirá imprimir, desde el dispositivo, las fichas previamente generadas. Pudiendo hacerlo de una en una o varias a la vez.
- Cada ficha de ejercicios equivaldrá a una hoja de papel e incluirá una sola operación matemática.
- La ficha impresa deberá incluir la operación matemática como una operación en vertical, pero excluyendo el resultado de esta.

RF05 Eliminar fichas de ejercicios.

- La aplicación permitirá borrar del dispositivo las fichas previamente guardadas.

3.1.1.2 Requisitos en la autoevaluación de resultados

RF06 Mostrar, mediante técnicas de realidad aumentada, la solución a las fichas de ejercicios.

- La aplicación deberá mostrar, con el uso de realidad aumentada, la solución a aquella ficha de ejercicios que este dentro de el punto de vista de la cámara del dispositivo.
- La solución deberá ser mostrada de forma virtual (realidad aumentada) al lado de la propia ficha y sin tapar la solución del propio usuario.
- La solución no debe aparecer en tres dimensiones, si no que deberá parecer que forma parte de la propia hoja o ficha.

RF07 Mostrar una pantalla de ayuda

- La aplicación deberá mostrar una pantalla que guíe al usuario en el uso de esta funcionalidad.

RF08 Cambiar la resolución de la cámara.

- La aplicación permitirá a los usuarios escoger la resolución de la cámara que mejor se adapte a las capacidades del dispositivo.
- Este ajuste deberá ser persistente, es decir, el valor se mantendrá entre sesión y sesión.

3.1.2 Requisitos no funcionales

A continuación, se describen los **requisitos no funcionales**:

RNF01 La aplicación deberá ser compatible con la plataforma de Android, independientemente de el tipo de dispositivo, *smartphone* o *tablet*.

RNF02 La interfaz gráfica debe ser simple, intuitiva y fácil de utilizar. De forma que cualquier tipo de usuario sepa usarla sin problemas.

RNF03 Todos los elementos y pantallas que conforman la interfaz gráfica deberán mantener un estilo y colores similares.

RNF04 La aplicación mostrará mensajes de error en caso de que la aplicación falle durante su ejecución. Estos mensajes deberán ser comprensibles para cualquier tipo de usuario.

RNF05 El tiempo de respuesta de la aplicación debe ser rápido. Si el usuario tuviese que esperar un largo tiempo durante sus acciones, se deberían mostrar elementos gráficos que adviertan al usuario de la espera.

RNF06 La aplicación debe ser desarrollada de tal forma que cualquier desarrollador pueda añadir nuevas funciones al código fuente de forma sencilla.

3.2 Análisis de las soluciones

Una vez descritos los requisitos de la aplicación, es fundamental llevar a cabo un pequeño análisis de aquellas soluciones existentes que nos permitan cumplir estos requisitos.

Tras analizar los requisitos, el mayor reto u objetivo que plantea este proyecto es la parte correspondiente al proceso de autoevaluación de los resultados por parte de los alumnos. Este proceso se puede dividir en dos subprocesos, los cuales implican los siguientes desafíos:

- En primer lugar, para que la aplicación sea capaz de mostrar la solución a una determinada ficha de ejercicios, esta deberá ser capaz, en primer lugar, de identificar mediante la cámara del dispositivo la ficha que el usuario intenta corregir. Lo cual quiere decir que cada ficha debería contener uno o mas elementos que la identifiquen. De forma que el dispositivo sea capaz de reconocerla correctamente.

- En segundo lugar, la aplicación debe ser capaz de mostrar de forma virtual, junto a la ficha de ejercicios física, la solución de esta. Esto implica que la aplicación debe rastrear la posición de la ficha en todo momento para establecer dónde y cómo mostrar la solución del ejercicio. Esto es, simplemente, realidad aumentada.

A continuación, se discutirán diferentes soluciones para ambos retos, se enumerarán tanto los pros como los contras de cada una de las propuestas y se establecerá un criterio de selección.

3.2.1 Soluciones para la identificación de fichas de ejercicios

Como se ha comentado anteriormente, para que la aplicación sea capaz de mostrar la solución de una determinada ficha de ejercicios, primero debe ser capaz de identificar, de alguna manera, dicha ficha de ejercicios.

Además, una vez identificada la ficha, la aplicación debe obtener, a partir de esta, la información de la solución, es decir, lo que debe mostrar. Por lo que también hay que decidir donde estará alojada esta información. Por ejemplo, podría estar almacenada en algún lugar externo (ej. una base de datos), podría estar almacenada implícitamente en la ficha (ej. información codificada) o, podría generarse dinámicamente a partir de la información que ya contiene la ficha (ej. ejercicios de operaciones matemáticas donde la solución se extrae de la lectura del operador y los operandos).

A continuación, se presentan las diferentes propuestas a estos problemas.

3.2.1.1 ROC

El ROC, o mas frecuentemente conocido por sus siglas en inglés (*ORC*), es el proceso conocido como Reconocimiento Óptico de Caracteres y consiste en identificar símbolos o caracteres de un determinado alfabeto dentro de una imagen.

Esta tecnología se podría llegar a utilizar en este proyecto dado que las fichas de ejercicios, como se describe en los requisitos, consistirán en simples operaciones matemáticas del tipo $X + Y$. Por lo que, mediante esta técnica, se podría identificar el operador y los operandos de la operación para calcular, dinámicamente, el resultado de esta.

Incluso, en un futuro, la aplicación podría ser capaz de identificar la solución escrita por el usuario, mediante la misma técnica, y determinar si es correcta o no.

Desafortunadamente, existen otros inconvenientes que hay que tener en cuenta. Uno de ellos, es que esta técnica no siempre es perfecta y la precisión depende mucho, entre otros factores, de la calidad de la imagen. Esta técnica funciona mejor con imágenes estáticas, en este caso, hay que tener en cuenta que el usuario estará "apuntando" con su dispositivo hacia la ficha de ejercicios por lo que habrá un movimiento constante que dificultará la identificación de los caracteres.

Otro gran inconveniente, es que esta técnica intentará identificar todos los caracteres o símbolos que reconozca en la imagen. Por lo tanto, cualquier otro texto en la ficha de

ejercicios, o fuera de ella, además del correspondiente a la operación matemática, podrían ser identificados sin desearlo, complicando la obtención de los caracteres necesarios.

3.2.1.2 Códigos QR

Un código QR, o código de respuesta rápida, es una evolución al conocido código de barras. Este, esta formado por una matriz de puntos blancos y negros que almacenan la información correspondiente, como texto, direcciones web, etc.

Entre sus ventajas, destacan por ser muy rápidos, fáciles de leer y por poder almacenar más de 4000 caracteres alfanuméricos [7].

Gracias a esto último, este tipo de códigos se podría usar para proporcionarle a la aplicación la información necesaria sobre una determinada ficha de ejercicios. En este caso, el tipo de operación matemática, los operandos implicados y el resultado de la operación. Por lo que la aplicación solo tendría que leer e interpretar este código para poder mostrar al usuario la solución de la ficha.

La otra solución es que el código QR, guarde simplemente un identificador de la solución y que la aplicación lo use para obtener la solución almacenada en una base de datos o similar. Pero de esta forma, la aplicación solo podría presentar al usuario, aquellas soluciones que ya existan previamente en la base de datos.

3.2.2 Soluciones para la realidad aumentada

Para poder hacer uso de la realidad aumentada, es necesario investigar todas las opciones disponibles en el mercado y escoger aquella que se adapte mejor a las necesidades de la aplicación cumpliendo con los requisitos de esta.

Para esta elección se ha establecido un primer filtro que consiste en dos requisitos básicos y obligatorios a cumplir. Estos requisitos son:

- ✓ Soporte para dispositivos Android (no se establece compatibilidad con una versión del SDK de Android mínima).
- ✓ Licencia de uso gratuita.

A continuación, se muestran todos los *SDKs* (orientados a la realidad aumentada) que han sido objeto de análisis. Se incluirán tanto las ventajas como las desventajas de cada uno de ellos y se aplicarán los requisitos anteriores para descartar o aceptar los que correspondan.

Así mismo, se recomienda la lectura del Anexo A para familiarizarse con algunos de los términos comúnmente utilizados en el ámbito de la realidad aumentada.

3.2.2.1 Vuforia

Vuforia es uno de los softwares mas populares y que mas tiempo lleva en el mercado. Por lo que también es uno de los mas avanzados y que mas funcionalidades ofrece. Entre sus ventajas están:

- Capacidad de reconocer y rastrear múltiples imágenes (*Image Targets*), tanto almacenadas localmente como remotamente.
- Reconocimiento en tiempo real de diferentes objetos en 3D (*Object Tracker*), como, por ejemplo, cajas o cilindros.
- Posicionamiento de objetos sin necesidad de marcadores, mediante *SLAM*.
- Reconocimiento de *VuMarks*, parecidos a los códigos QR, pero combinables con imágenes para hacerlos más personalizables.
- Compatibilidad con las plataformas Android, iOS, UWP y Unity.
- API compatible con los lenguajes de programación C++, Java y Objective-C.

Todas estas ventajas cumplen completamente con los requisitos de la aplicación, pero desafortunadamente, es un software de pago pensado para proyectos y compañías más grandes. Ofrecen una licencia gratuita para desarrollo y pruebas, pero no es para uso comercial e incluye una marca de agua.

3.2.2.1 Wikitude

Wikitude comparte muchas de las características con Vuforia, pero, en este caso el desarrollo se hace a través de los *frameworks* de Cordova, Appcelator Titanium, Xamarin o Unity. Por lo que lo hace compatible con una gran variedad de dispositivos iOS, Android, etc.

Es, además, una especie de todo en uno, ya que además de proporcionar su propio *SDK*, este añade a su vez, soporte para los *SDKs* ARKit de Apple y ARCore de Google. Lo que resulta en una combinación de los tres.

Lamentablemente, al igual que Vuforia, es de pago. La licencia gratuita sólo incluye un periodo de prueba para experimentar con la herramienta.

3.2.2.2 EasyAR

EasyAR destaca por incluir una versión completamente gratuita de su *framework*.

Es compatible con una gran variedad de dispositivos y lenguajes de programación (C, C++, Java, Kotlin, Objective-C y Swift) y, además, también añade reconocimiento de códigos QR.

Ofrece dos tipos de licencias, una gratuita y otra de pago. La diferencia entre ellas es que la primera pierde algunas funcionalidades, como, por ejemplo, la funcionalidad de *SLAM* y el reconocimiento de objetos 3D. Pero, al contrario que las anteriores, esta no incluye ningún tipo de marca de agua.

3.2.2.1 ARToolKit

ARToolKit se caracteriza por ser completamente *open-source*, es de código libre y fue desarrollada inicialmente por Hirokazu Kato en 1999.

Al ser *open-source*, se pueden encontrar una gran cantidad de variantes y proyectos que surgieron de esta, como por ejemplo NyARToolkit. El problema de estas variantes es que la mayoría están descontinuadas o no poseen demasiada documentación.

3.2.2.2 ARCore

ARCore destaca por ser una plataforma de AR desarrollada directamente por Google. Por ello, por ser gratuita y por las funcionalidades que ofrece, la convierten en una buena opción en el desarrollo de aplicaciones de realidad aumentada.

La desventaja es que, actualmente (en el momento de desarrollo de este proyecto), esta sólo soporta versiones de Android muy recientes (7.0 en adelante) y aún así, no todos los dispositivos que se encuentran en esa versión son completamente compatibles.

3.3 Solución propuesta

Los dos dilemas mas importantes encontrados durante la fase de análisis fueron, el reto de la identificación de las fichas de ejercicios y el reto de la realidad aumentada. Tras haber analizado las diferentes propuestas en los apartados 3.2.1 y 3.2.2, correspondientemente, se describe, a continuación, la solución más conveniente para el desarrollo de esta aplicación.

En primer lugar, de entre las librerías de realidad aumentada analizadas. Podemos descartar, a pesar de sus muchas funcionalidades, los *SDKs* de Vuforia y Wikitude. Ya que, como hemos comentado, estos son de pago y están orientados a proyectos de mayor magnitud.

De entre las restantes (EasyAR, ARToolKit y ARCore), todas son gratuitas u ofrecen versión gratuita y son compatibles con dispositivos Android. Por lo tanto, la decisión ha sido tomada evaluando otros aspectos y características.

ARCore es la que menos dispositivos compatibles soporta, y a pesar de que los requisitos de la aplicación no especifican una versión de compatibilidad de Android mínima, se busca que la aplicación soporte el mayor número de dispositivos posibles.

Por otro lado, dado que parece que la versión original de ARToolKit ya no recibe soporte y el nivel de documentación de esta y de otros proyectos relacionados no es muy alta, se ha determinado que la herramienta mas adecuada para este trabajo es el *framework* de EasyAR.

La versión gratuita de EasyAR, ofrece todas las funcionalidades de realidad aumentada necesarias, es un proyecto relativamente grande, esta bien documentado y ofrece pequeños proyectos de ejemplo, a modo de guía, para el desarrollo.

Por último, la solución más sencilla a la identificación de fichas de ejercicios y que mejor se adapta con la otra solución, es el uso de códigos QR. Dada su capacidad de almacenamiento, se pueden usar para guardar implícitamente la información del ejercicio y mostrar la solución a este dinámicamente, sin necesidad de realizar consultas a bases de datos o similar.

Por lo tanto, cada código QR contendrá la información codificada de la operación matemática en forma de texto (ej. "3+5=9"). Esto texto será interpretado por la aplicación cuando el código sea leído y se pintará en el formato adecuado.

De esta manera, se usará la librería **EasyAR** tanto como ayuda en el uso de realidad aumentada, como para la identificación de fichas mediante la lectura de códigos QR.

4 Diseño y desarrollo

En este apartado se definirá y presentará el diseño y la arquitectura general de la aplicación a partir de las soluciones y requisitos definidos en la sección anterior.

En primer lugar, se describirán las herramientas con las que se va a implementar la solución, esto es, entorno, lenguajes, librerías, *APIs*, etc. Seguidamente, se detallará la arquitectura software del sistema, es decir, la descripción de los componentes internos, que hacen y cómo se comportan e interactúan entre ellos.

4.1 Análisis de las herramientas

En las siguientes dos subsecciones, se detallarán los entornos de desarrollo y lenguajes de programación que se tuvieron en cuenta antes del desarrollo, así como la elección de estos.

Además, se describirán las diferentes librerías y *frameworks* utilizados durante la realización del proyecto.

4.1.1 Entorno

Actualmente, existe una gran variedad de entornos de desarrollo o *IDEs* (***Integrated Development Environment***) que soporten el desarrollo de aplicaciones para dispositivos Android.

Un ***IDE*** es simplemente un software o aplicación informática que proporciona una serie de servicios, herramientas y componentes que facilitan el desarrollo software, aumentando así la productividad del programador.

Figura 4-1: Entornos de desarrollo integrado para Android

En el caso de Android, entre los mas populares están: **Android Studio**, **IntelliJ IDEA**, **Eclipse** y **Netbeans** [Figura 4-1]. Todos son gratuitos u ofrecen una versión gratuita. Antes de que Android Studio existiera, el IDE oficial para el desarrollo de aplicaciones Android era Eclipse. En mayo de 2013, Google anunció Android Studio (basado en IntelliJ IDEA) y reemplazó a Eclipse como la herramienta de desarrollo oficial. Desde entonces, se ha ido convirtiendo en, prácticamente, el estándar.

Debido a que esta diseñado específicamente para el desarrollo de Android, esta soportado oficialmente por Google, por experiencia anterior con él y otras ventajas que ofrece, se ha escogido **Android Studio** como entorno de desarrollo para este proyecto.

4.1.2 Lenguajes

Como con el entorno de desarrollo, existen también varias opciones a la hora de escoger un lenguaje de programación compatible con el desarrollo para Android.

Aunque la elección de este lenguaje estará también influenciada por el entorno de desarrollo escogido, ya que es posible que un determinado entorno no sea compatible con un determinado lenguaje.

En este caso, Android Studio (escogido en la sección anterior), te permite escoger entre dos lenguajes de programación, **Java** y **Kotlin**. Este último es relativamente nuevo en el desarrollo para Android. Google anunció soporte oficial a este lenguaje a mediados de 2017 y se convirtió en una alternativa al desarrollo en Java.

Además de Java y Kotlin, Google también da soporte, mediante la herramienta Android NDK, a los lenguajes **C** y **C++**. Haciendo que sea posible implementar parte del código mediante estos lenguajes, o a su vez, utilizar librerías que hayan sido escritas en estos lenguajes.

Para este proyecto, no se prevé el uso de ninguno de estos dos últimos lenguajes (C y C++) ya que los requisitos y funcionalidades pedidas no lo requieren.

Por último, entre Java y Kotlin se ha escogido el uso de **Java** simplemente por conocimientos y grado de experiencia.

4.1.3 Librerías

4.1.3.1 Android Support Library

Android Support Library es una colección de librerías las cuales ofrecen una serie de funcionalidades que no están incluidas por defecto en el *framework* de Android. Entre algunas de sus características cabe mencionar las siguientes:

- Permiten usar componentes y funcionalidades de versiones nuevas de Android en versiones mas antiguas.
- Ofrecen elementos de la interfaz gráfica adicionales. Por ejemplo, *RecyclerView* para el uso de listas, o elementos de *Material Design,* como los *Floating Action Buttons*.

4.1.3.2 Android Architecture Components

Se trata de una colección de librerías destinadas a ayudar a los desarrolladores a diseñar aplicaciones mas robustas, fáciles de mantener y testeables. Entre las cuales, se han usado las siguientes:

- **ViewModel** - Es una clase que esta diseñada para almacenar y administrar datos relacionados con la interfaz de usuario. La ventaja es que esta clase no sufre los cambios de configuración de las Activities (como las rotaciones de pantalla). También, manejará la comunicación de la Actividad/Fragmento con el resto de la aplicación, por ejemplo, se encargará de llamar a las clases que contienen la lógica de negocio. Es, en resumen, una forma útil de seguir uno de los principios básicos de la programación orientada a objetos, la separación de responsabilidades.
- **LiveData** - Es una clase para almacenar datos que sigue el patrón de diseño *Observer pattern*, en el cual, los "observadores" son notificados automáticamente cuando el estado del objeto "observado" cambia.
- **Room** - Es una librería que proporciona una capa de abstracción sobre la implementación de bases de datos SQLite. Haciendo mas sencilla la implementación este tipo de bases de datos. Además, es compatible con la librería anterior, LiveData.

4.1.3.3 EasyAR

Como se ha determinado en el capítulo 3.3, la librería usada para las funciones de realidad aumentada va a ser EasyAR. Esta se usará para rastrear y determinar la posición de las fichas de ejercicios en la vida real y, además, se utilizará como lector de códigos QR.

Dado que esta librería necesita una imagen o marcador para poder determinar la posición del objeto final. Se ha diseñado un marcador que permita ser fácilmente distinguible, que vaya acorde con la temática de la aplicación y que además permita incluir en su interior un código QR.

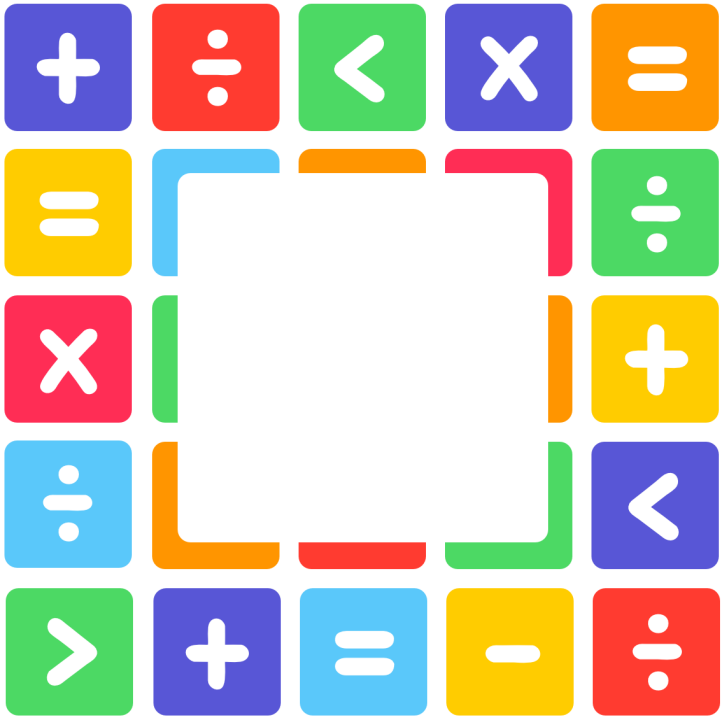

Figura 4-2: Marcador AR de la aplicación

4.1.3.4 OpenGL ES

OpenGL ES, es una variante de OpenGL diseñada para dispositivos integrados. En este caso se usará la versión 2 de esta herramienta y permitirá el renderizado de gráficos 2D y 3D. Se utilizará para renderizar el objeto 3D correspondiente a la solución de una determinada ficha de ejercicios, el cual será mostrado en forma de realidad aumentada.

4.1.3.5 ZXing

Se trata de una librería *open-source* la cual será utilizada para generar códigos QR. Esta se usará a la hora de imprimir fichas de ejercicios para generar el código QR correspondiente a la ficha que se quiere imprimir.

4.2 Arquitectura software

En esta sección se describe la lista de clases y los principales elementos que toman parte durante la ejecución de la aplicación. Se especificarán las relaciones que mantienen, su estructura y su funcionalidad.

Para explicar la arquitectura de la aplicación, es importante primero, observarla empezando desde un punto de vista de alto nivel. Si recordamos los objetivos principales de la aplicación, descritos durante la fase de análisis, podíamos dividir la aplicación en dos funcionalidades principales. La primera es el uso de la realidad aumentada para la autoevaluación de fichas de ejercicios y la segunda es la gestión de estas fichas de ejercicios.

El siguiente diagrama de clases simplificado muestra las principales clases de la aplicación:

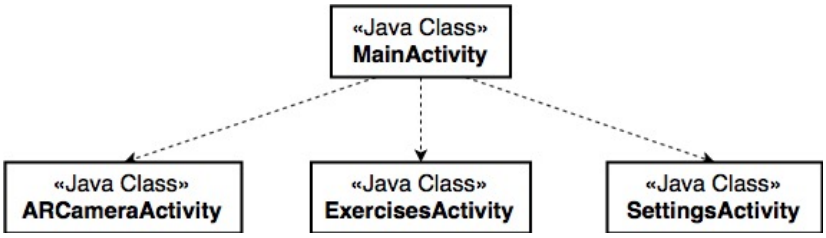

Figura 4-3: Diagramas de clases simplificado

Cada una de estas clases representa una vista de la aplicación. La clase ***MainActivity***, por ejemplo, se corresponde con la pantalla principal de la aplicación. Por otro lado, las clases ***ARCameraActivity*** y ***ExercisesActivity*** se corresponden con las dos funcionalidades principales comentadas anteriormente. La primera se corresponde con la vista de la aplicación que proporcionará la funcionalidad de la realidad aumentada (para la autoevaluación de los ejercicios) y la segunda se corresponde con la vista desde la que se podrán gestionar las fichas de ejercicios. Por último, se ha añadido la clase ***SettingsActivity***, la cual contiene la vista de las preferencias o ajustes de la aplicación.

El diagrama muestra también, mediante flechas, las relaciones o dependencias entre estas clases. En este caso se usan simplemente para mostrar que las clases dependientes, es decir, *ARCameraActivity*, *ExercisesActivity* y *SettingsActivity* son invocadas o llamadas por la clase de la que dependen, esto es, *MainActivity*.

Una vez descritas las principales clases o vistas de la aplicación, podemos describir con más detalle las dos más importantes.

4.2.1 Arquitectura en la realidad aumentada

A continuación, se describe la arquitectura de las principales clases implicadas en la funcionalidad de la realidad aumentada durante la autoevaluación de las fichas de ejercicios.

Esta se compone de cuatro clases principales:

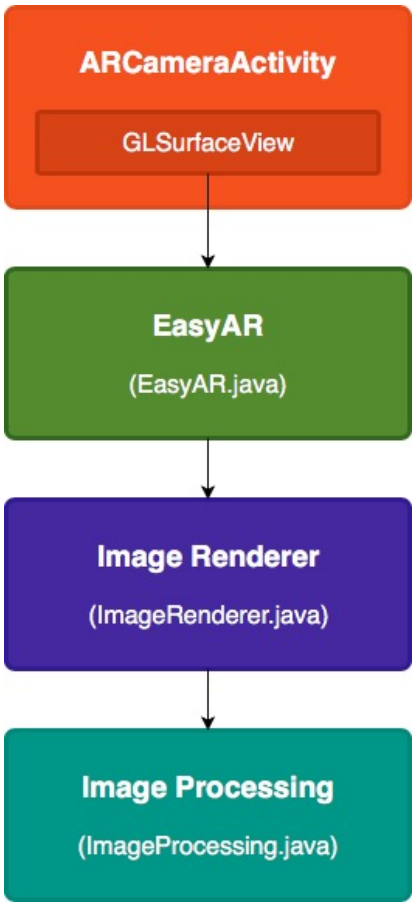

Figura 4-4: Arquitectura de las clases implicadas en la realidad aumentada

- **ARCameraActivity** - Es la pantalla principal de la cámara. Contiene una vista llamada GLSurfaceView, la cual es un tipo de vista especial que permite pintar objetos mediante la librería OpenGL ES.
- **EasyAR** - Es la clase que contiene todas las funcionalidades que proporciona la librería EasyAR. Como el seguimiento de imágenes o marcadores de AR y la lectura de códigos QR a través de la cámara del dispositivo.
- **Image Renderer** - Esta clase contiene los métodos que se encargan de el renderizado de objetos 3D mediante la librería OpenGL ES. Es llamada constantemente por la clase EasyAR, la cual le proporciona la información en todo momento de la posición de la ficha en el entorno para poder pintarla en la posición correcta. Cada vez que la clase EasyAR detecte un nuevo código QR, esta clase llamará a la clase ImageProcessing para que le genere una nueva imagen que renderizar.
- **Image Proccessing** - En esta parte de la arquitectura, esta clase es usada por la clase ImageRenderer para obtener una imagen (*Bitmap*) representando la solución que se debe mostrar. Esta imagen será generada a partir de la información extraída de el código QR leído.

4.2.2 Arquitectura en la gestión de fichas

A continuación, se muestra la arquitectura de las principales clases implicadas en la funcionalidad de la gestión de fichas de ejercicios.

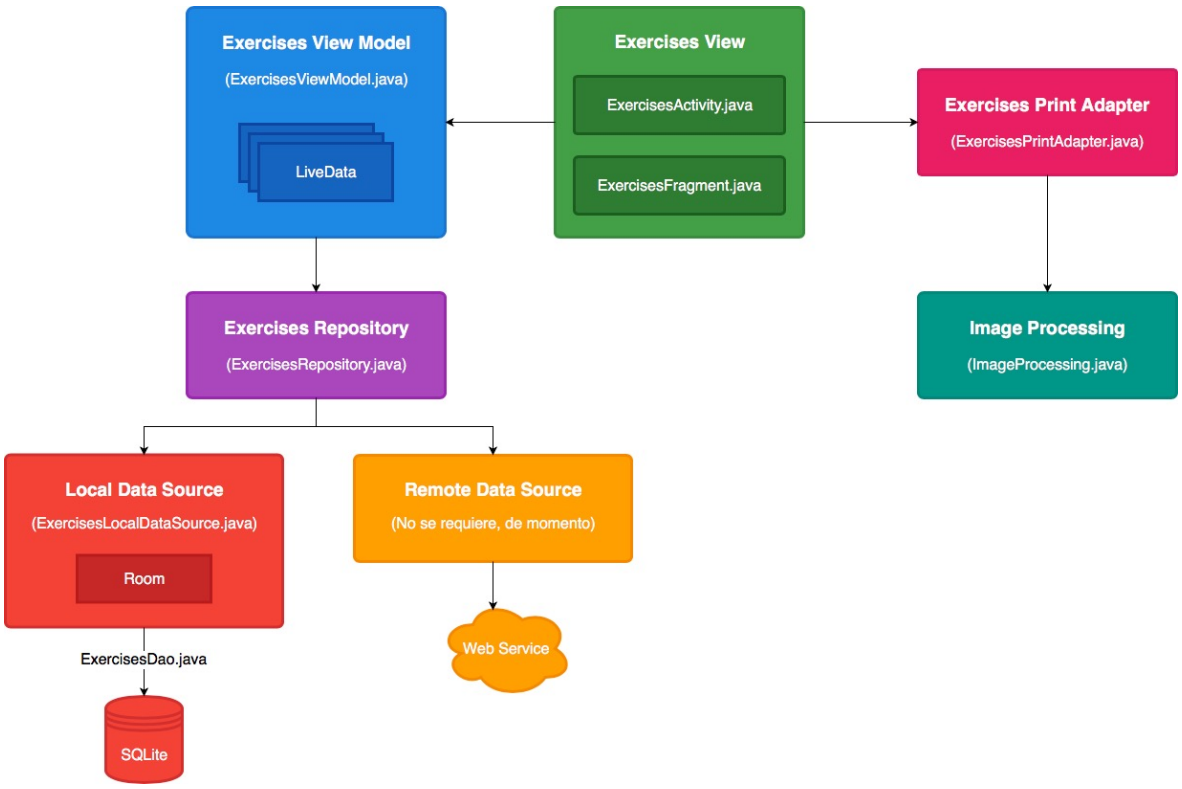

Figura 4-5: Arquitectura de las clases implicadas en la gestión de fichas

Esta arquitectura se basa en el patrón de arquitectura software **MVVM** (Model-View-ViewModel) y en el patrón de **repositorio** (Repository Pattern).

Este el diseño recomendado por Google para el uso de los Android Architecture Components [2], que ya hemos visto en la sección 4.1.3.2. De esta forma se consigue separar de forma efectiva la interfaz de usuario de la lógica de la aplicación.

También, se ha decidido usar el patrón de repositorio para la parte de acceso a los datos. De esta forma el *ViewModel* (*Exercises View Model*) no necesita saber con que tecnología o de que manera se accede a los datos, sino que solo conoce las operaciones que le facilita el repositorio para el acceso a estos datos. Un repositorio puede esta formado por varias fuentes de datos (*Data Sources*). En el caso de esta aplicación, el repositorio (*Exercises Repository*) sólo esta formado por una fuente de datos local (*Local Data Source*), debido a que ésta no requiere ninguna otra fuente de datos a parte de la base de datos local. Aún así, implementando este patrón facilitamos que en un futuro se puedan añadir de forma muy sencilla, por ejemplo, un servidor de datos remoto (*Remote Data Source)*.

Por otro lado, la clase **ExercisesPrintAdapter** se encarga de imprimir las fichas de ejercicios seleccionadas por el usuario. Para ello, hace uso de la clase ImageProcessing, la cual ya hemos visto en la sección anterior. En este caso, en vez de generar la imagen correspondiente a la solución de la ficha, esta genera una nueva ficha de ejercicios, la cual contendrá el ejercicio sin resolver, el marcador de realidad aumentada y el código QR con la información del ejercicio correspondiente. Se puede ver el ejemplo de esta ficha en el Anexo D.

4.3 Modelo de datos

Los datos que maneja la aplicación correspondiente a las fichas de ejercicios, es muy simple.

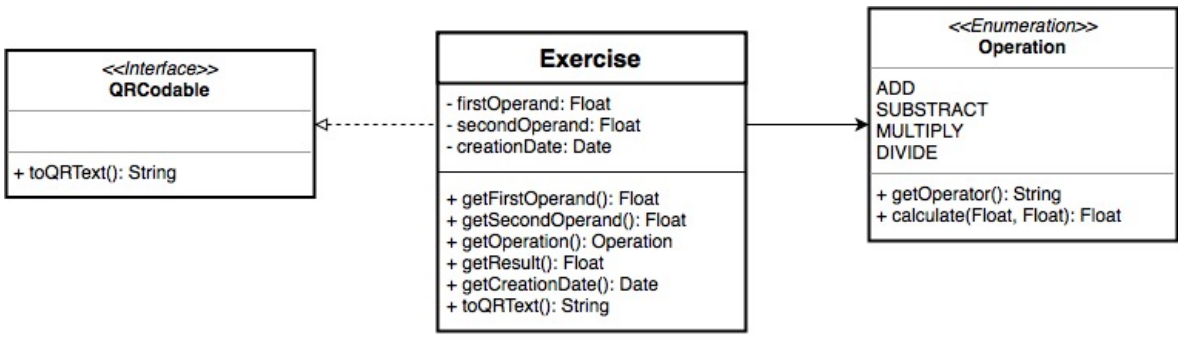

Figura 4-6: Modelo de datos

Un ejercicio (o ficha de ejercicios), no es mas que una operación matemática. La cual esta formada por dos operandos (***firstOperand*** y ***secondOperand***) y un tipo de operación (***Operation***). Este tipo de operación es una Enumeración en Java y puede ser una suma, una resta, una multiplicación y una división.

Además de estos tres atributos, un ejercicio también contiene la fecha en la que fue creado (***creationDate***), se usa simplemente para mostrar información adicional al usuario.

Por otro lado, cada ejercicio implementa el método ***toQRText()***, de la interfaz ***QRCodable***. Esto permite que un ejercicio pueda ser codificado en un código QR. Por lo tanto, cuando la aplicación imprima un ejercicio, el código QR generado contendrá la información obtenida de esta función.

Este texto codificado es simplemente una combinación de los atributos del ejercicio, de tal forma que, si un ejercicio tiene como primer operando un 5, como segundo operando un 22 y como tipo de operación una suma, el texto resultante sería: “5+22=27”.

5 Pruebas y resultados

En esta sección se describirá el trabajo realizado durante la fase de pruebas del proyecto. Se expondrán los diferentes tipos de pruebas realizadas y se analizarán los resultados de estas.

5.1 Tipos de pruebas

La mayoría de las pruebas realizadas durante el desarrollo del proyecto han sido **pruebas funcionales**, es decir, basadas en la ejecución y verificación de los requisitos funcionales de la aplicación.

Para comenzar, a medida que se iban implementando funcionalidades durante la fase de desarrollo, se iban desarrollando **pruebas unitarias**. Estas simplemente comprueban el correcto funcionamiento de pequeñas unidades de código, es decir, verifican que el código hace lo que tiene que hacer.

A parte de las pruebas unitarias, una vez que alguna funcionalidad o requisito se daba por terminado, se realizaban **pruebas de integración**, esto es para comprobar que todas las pequeñas unidades de código y métodos que forman parte de esa funcionalidad se comportan correctamente cuando se ejecutan conjuntamente.

Además de pruebas funcionales, también se han realizado algunas **pruebas no funcionales**, como, por ejemplo, **pruebas de compatibilidad**. Estas se han realizado tanto con diferentes tipos de dispositivos, *smartphones* y *tablets*, así como con diferentes versiones del sistema operativo Android, en concreto, con versiones entre la 5.0 y la 7.0.

Desafortunadamente, debido al tiempo y a la carga de trabajo no se han podido realizar pruebas con los usuarios finales, por lo tanto, estas pruebas de aceptación se propondrán como línea de trabajo futuro.

5.2 Resultados

A pesar de que la aplicación no ha podido ser testeada con los usuarios finales de la aplicación, esta ha sido sometida constantemente a diferentes tipos de pruebas durante todo su desarrollo. Desde el punto de vista del equipo de desarrollo se considera que la aplicación ha sido capaz de reunir todos los requisitos especificados, pero será el usuario final quien deba decidir con que grado de aceptación se han cumplido.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Este proyecto tenía como objetivo principal el desarrollo de una aplicación móvil para ayudar a alumnos con diversidad funcional intelectual a autoevaluar sus resultados gracias al uso de la realidad aumentada. Y a suministrar a los profesores una herramienta para la generación de fichas de ejercicios compatibles con esta tecnología.

Durante estos meses se han puesto de manifiesto y se han integrado los conocimientos aprendidos durante la carrera, pero también se han ganado muchos otros. He tenido la oportunidad de trabajar en un proyecto que me ha permitido explorar el uso de nuevas tecnologías (como la realidad aumentada) y nuevas áreas de conocimiento.

Además, creo que esta herramienta tiene una gran finalidad, y posee un gran potencial.

Al final, se ha conseguido completar en mayor o menor la medida todas las fases de este proyecto pudiendo así satisfacer los objetivos del proyecto.

6.2 Trabajo futuro

El primer punto de trabajo futuro y uno de los mas importantes, sería centrar el objetivo de evaluación de la aplicación sobre los usuarios finales. Se deberían realizar pruebas de aceptación y de cumplimiento de requisitos. Además de ayudar a determinar el cumplimiento de los objetivos, aportaría una buena retroalimentación que ayudaría a determinar mas posibles líneas de futuro.

En cuanto ha funcionalidades o mejoras sobre la aplicación se proponen las siguientes:

- Poder autoevaluar y crear mas tipos de ejercicios, no solo mas tipos de ejercicios matemáticos, si no también ejercicios enfocados en otras áreas de la educación.
- Mejoras en el visualizado de las soluciones mediante realidad aumentada, como, por ejemplo, con objetos en tres dimensiones o soluciones de ejercicios animadas, que expliquen la realización del ejercicio.
- La posibilidad de guardar las fichas de ejercicios en la “nube”, para que puedan ser accedidas remotamente, pare que puedan ser compartidas, quizás, entre profesores de un mismo colegio.
- Añadir compatibilidad con mas plataformas, por ejemplo, iOS.
- A más largo plazo, la aplicación podría incluso evaluar automáticamente a los alumnos (en algunos tipos de ejercicios) y compartir esta evaluación con los profesores.

Referencias

[1] I. Sommerville, «Software Engineering» 9th ed., Pearson, 2010, pp. 84-87.

[2] Google LLC, «Android Architecture Components» [En línea]. https://developer.android.com/topic/libraries/architecture/.

[3] Arloon, «Aplicaciones de Arloon» [En línea]. http://www.arloon.com.

[4] PBS Kids, «FETCH! Lunch Rush» [En línea]. http://pbskids.org/apps/fetch-lunch-rush.html.

[5] QuiverVision, «Quiver Education» [En línea]. http://www.quivervision.com/apps/quiver-education/.

[6] Aurasma, [En línea]. https://www.aurasma.com.

[7] Wikipedia, «Código QR,» [En línea]. https://es.wikipedia.org/wiki/Código_QR.

Glosario

API	Application Programming Interface
AR	Augmented Reality
IDE	Integrated Development Environment
MVVM	Model-View-ViewModel
NDK	Native Development Kit
OCR	Optical Character Recognition
QR	Quick Response
ROC	Reconocimiento Óptico de Caracteres
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
TIC	Tecnologías de la Información y la Comunicación
UWP	Universal Windows Platform

Anexos

A Términos en la realidad aumentada

Las técnicas en el uso de realidad aumentada se pueden dividir en dos tipos:

- **Marker-Based** - Es aquella que esta basada, básicamente, en el reconocimiento de una imagen. Esta imagen o marcador suele ser un cierto patrón o imagen distinguible. Cuando es reconocida, el sistema superpone en la posición de dicha imagen el modelo u objeto que se quiere mostrar. La orientación de este objeto dependerá de la posición del marcador.
- **Marker-Less -** Es aquella que no usa ningún tipo de imagen o marcador para posicionar el objeto a mostrar en el entorno, sino que suelen utilizar una tecnología conocida como **SLAM** que crea una especie de mapa virtual a partir de una nube de puntos sobre el espacio.

Otras definiciones:

- **Object Tracker** - Se refiere a la capacidad para reconocer y rastrear la posición objetos en 3D, haciendo posible la superposición de un objeto 3D virtual sobre él.
- **Image/Marker Tracker** - Es la capacidad de reconocer y rastrear la posición de una determinada imagen o marcador en el entorno.

B Manual de usuario

Figura 0-1: Pantalla de inicio de la aplicación

La pantalla de inicio cuenta con tres botones:

- **"Corrige tus ejercicios"** - Para la evaluación de las fichas de ejercicios.
- **"Ejercicios"** - Para la gestión de fichas de ejercicios.
- **"Ajustes"** - Para modificar las preferencias de la aplicación.

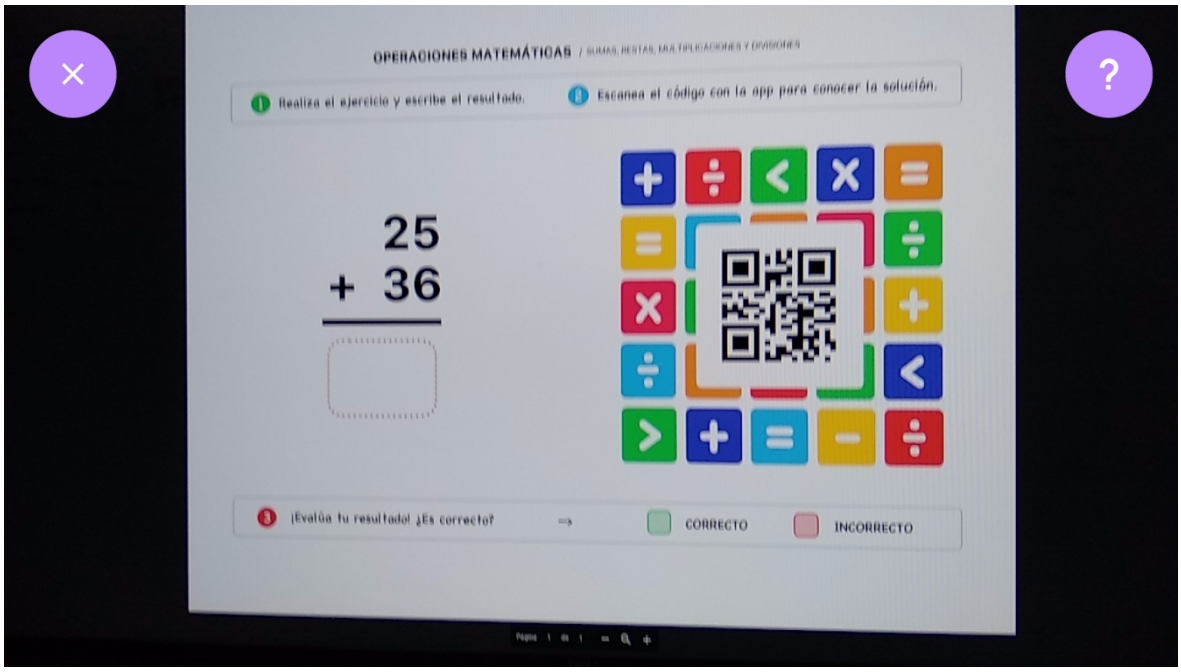

Figura 0-2: Pantalla de la aplicación para la corrección de ejercicios

La pantalla correspondiente a la corrección de ejercicios solo contiene dos botones, un botón para salir (a la izquierda) y un botón de ayuda (a la derecha).

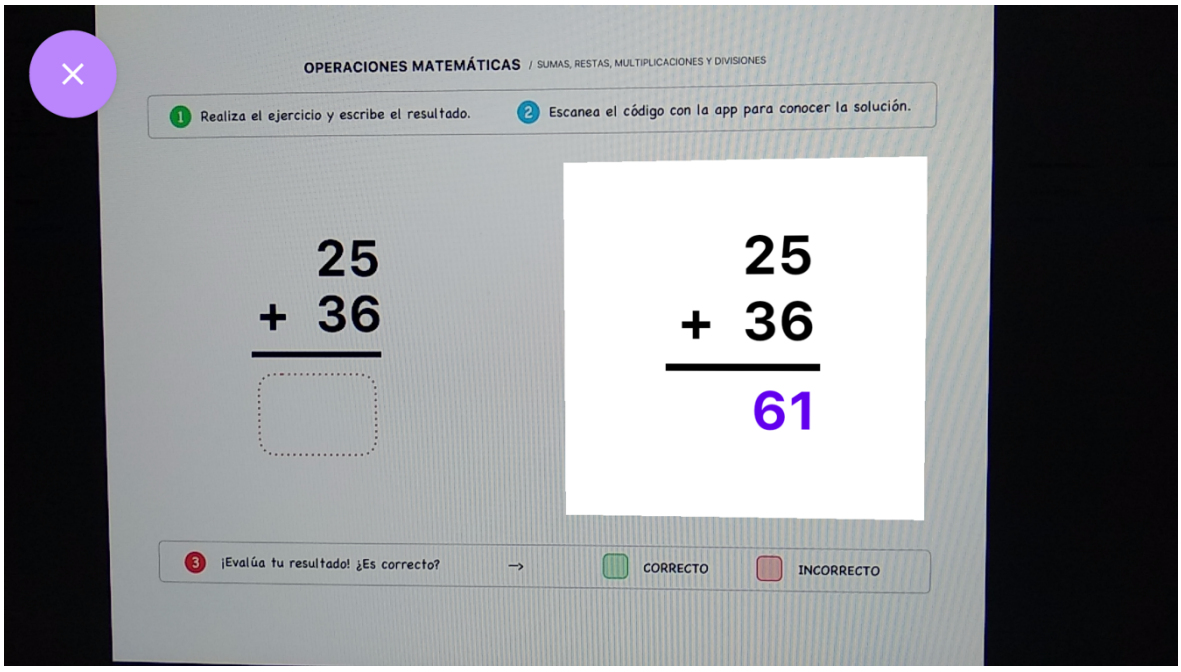

Figura 0-3: Pantalla de la aplicación para la corrección de ejercicios (con AR)

Cuando la aplicación detecta la imagen del marcador y el código QR, se muestra la solución al ejercicio en forma de realidad aumentada.

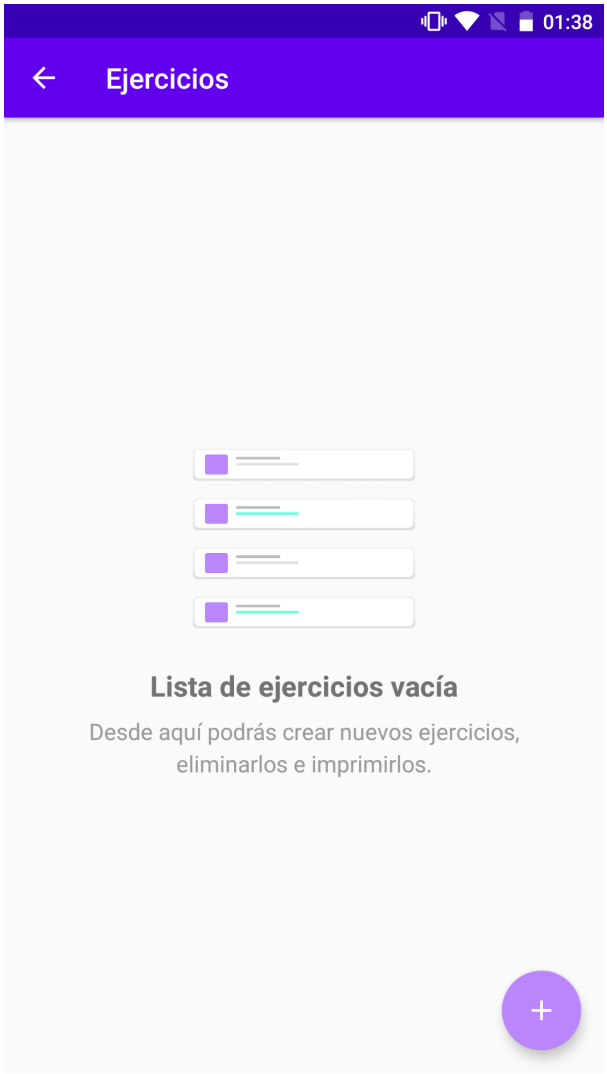

Figura 0-4: Lista de ejercicios vacía

La primera vez que se abre la aplicación la lista de ejercicios estará vacía. Para añadir un ejercicio se debe pulsar el botón con el símbolo '+' de abajo a la izquierda.

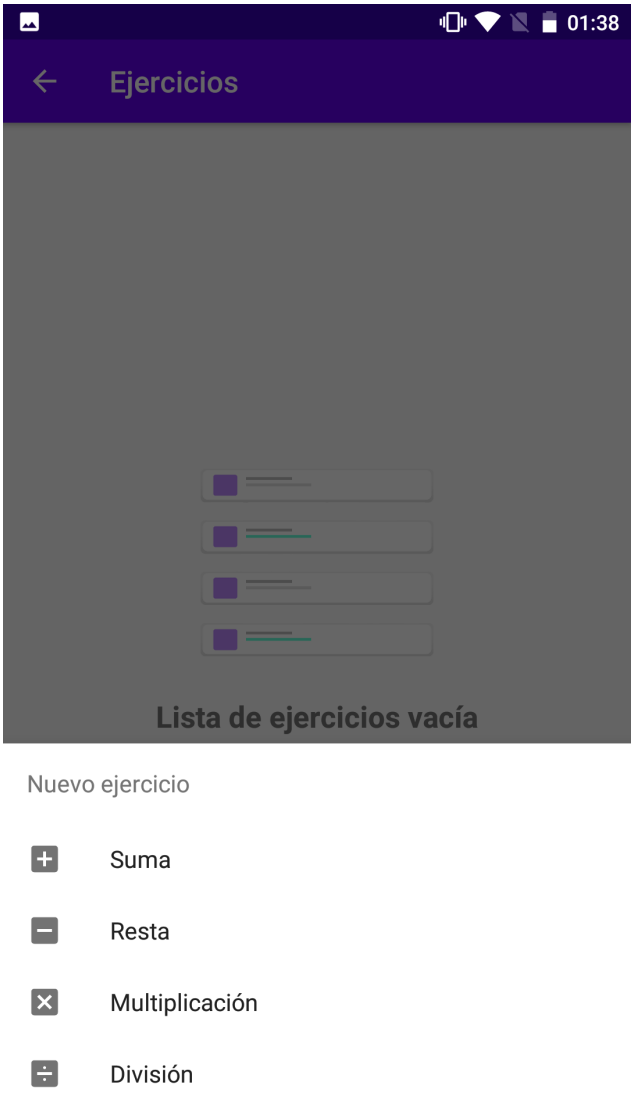

Figura 0-5: Tipos de ejercicios

Una vez pulsado el botón de añadir ejercicio, se mostrará un menú con el tipo de ejercicio que se quiere crear.

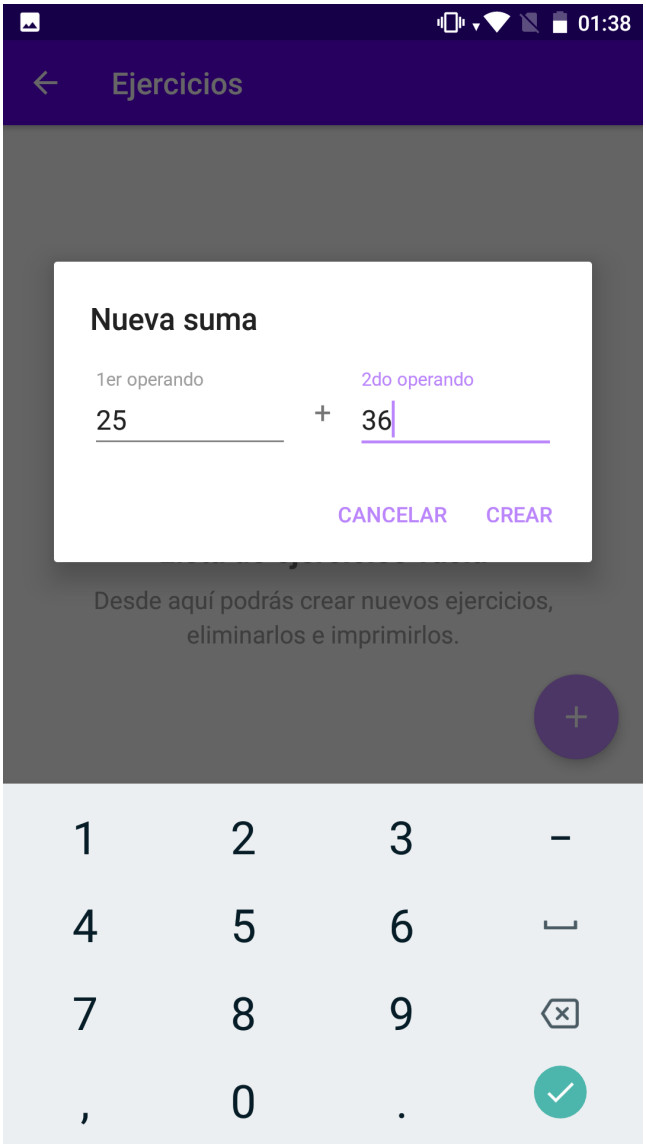

Figura 0-6: Nuevo ejercicio de tipo suma

Una vez escogido el tipo de ejercicio, solo queda configurar el valor de los operandos.

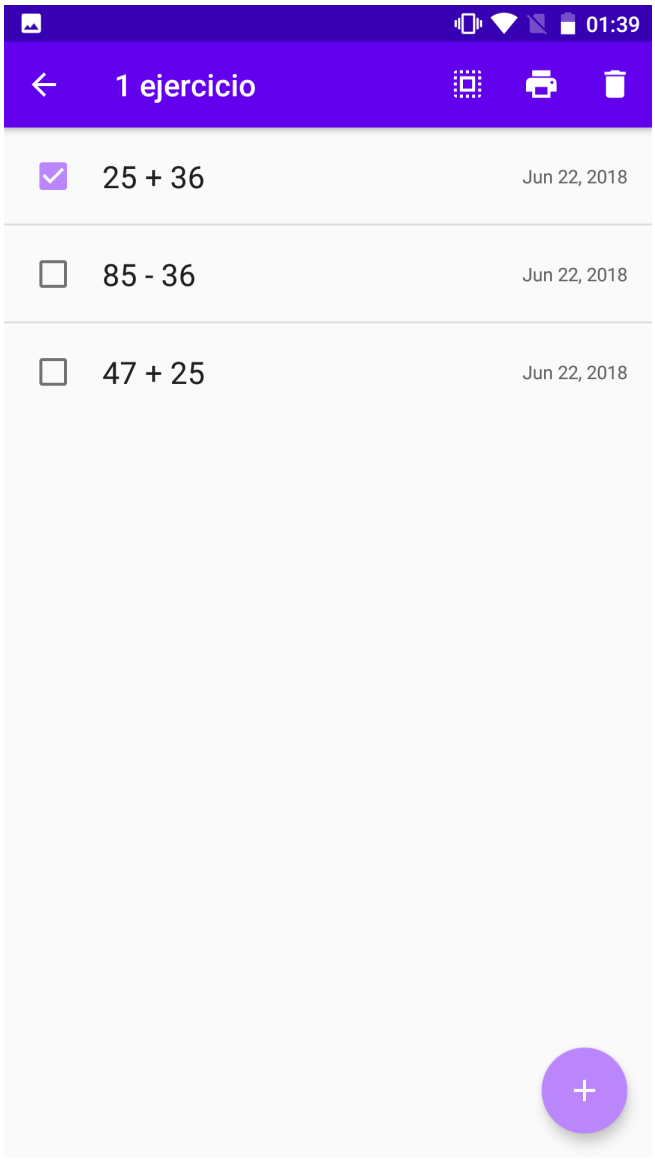

Figura 0-7: Lista de ejercicios

Una vez que se ya se han añadido ejercicios, estos aparecerán en una lista. Cada una de las filas (ejercicio) contiene un *checkbox* a su izquierda. Al pulsar sobre estos se mostrará un menú de opciones (arriba a la derecha). Desde este menú podremos seleccionar todos los ejercicios (primera opción), imprimir los ejercicios seleccionados (segunda opción) o eliminar los ejercicios seleccionados (tercera opción).

C Maquetas de la aplicación

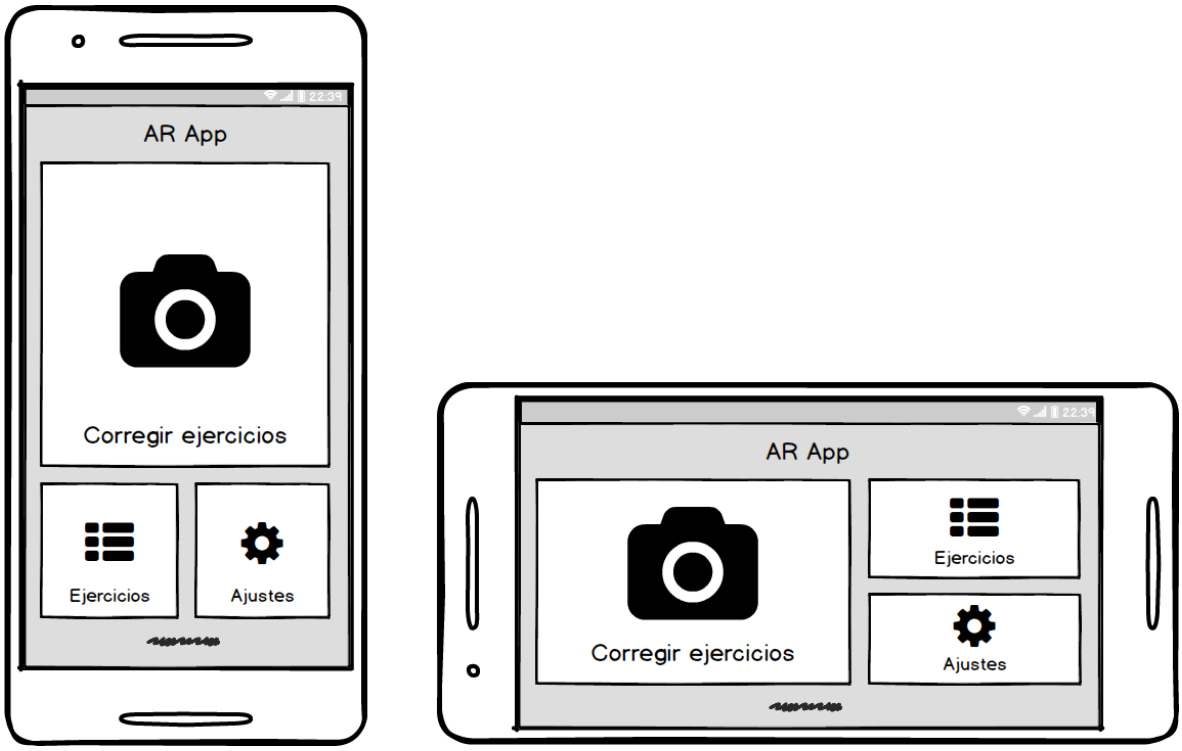

Figura 0-8: Maqueta de la pantalla de inicio

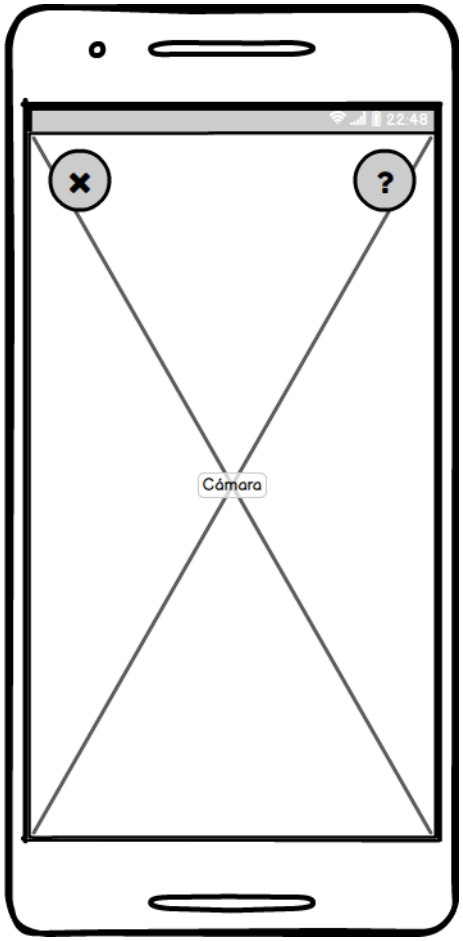

Figura 0-9: Maqueta de la vista de realidad aumentada

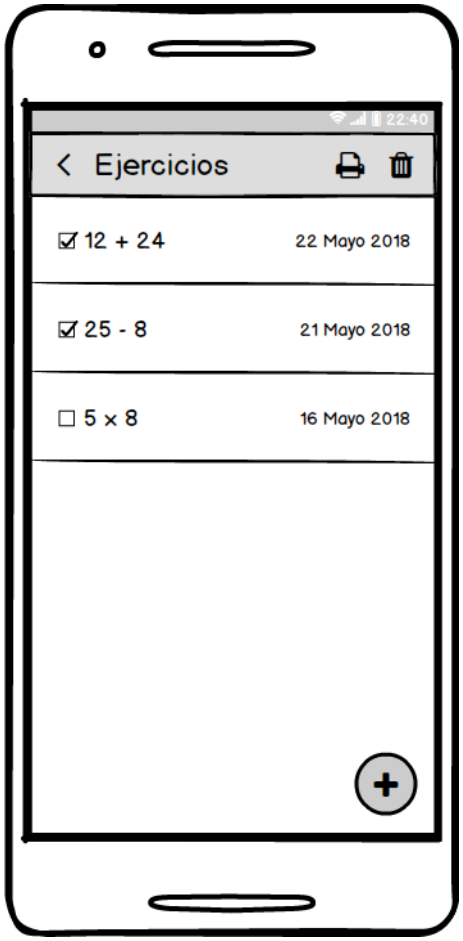

Figura 0-10: Maqueta de la gestión de ejercicios

D Ejemplo de ficha de ejercicios

Figura 0-11: Ejemplo de ficha de ejercicios